Regular Expressions

.	Any single character
*	0 or more of the preceding character
+ \+	One or more of the preceding character
? \?	0 or 1 of the preceding character
{ } \{ \}	Match exactly num of the preceding character: {x}- match x repetitions {x,} match x or more {,x} match 0 – x repetitions {x,y} match x to y repetions
^	Match Beginning of line
$	Match End of line
< \<	Match Beginning of word
> \>	Match End of word
[ ]	Range – match any character within the range
[^]	Negation – match any character NOT within the range
( ) \(\)	Group a pattern as a single entity
\1 ... \9	Match group \#
| |	Separate choices to match (| is the c-language token for logical OR)
The following have special meaning only in replacement patterns	
&	Reuse the search pattern as the current replacement pattern
\u	Convert first character of replacement pattern to uppercase
\U	Convert entire replacement pattern to uppercase
\l	Convert first character of replacement patter to lowercase
\L	Convert entire replacement pattern to lowercase

vi Commands

:w	Write an already named file to disk.
:w *filename*	Save an unnamed document or save as a new name.
:w! *filename*	Required to overwrite an existing file other than the one being edited.
:q	Quit vi.
:q!	Quit without saving.
:wq	Save document and quit.
:e *filename*	Open a file for editing or begin a new named doc.
:r *filename*	Add contents of file *filename* at cursor.
h	Move left one character
j	Move down one character
k	Move up one character
l	Move right one character
u	Undo
[Ctrl] r	Redo
dd	Delete current line
x or [Del]	Delete current character
yy or Y	Yank (copy) current line
p	Paste text between current line and next line
P	Paste text between current line and previous
/*text*	Find the next instance of *text*
:s/*old*/*new*	Replace next instance of *old* with *new*
:%s/*old*/*new*	Replace all instances of *old* with *new*
:help	Display online help

Common Commands

ls	List files (directory listing)
cp	Copy files
mv	Move (rename) files
rm	Remove (delete) files
mkdir	Make directory
rmdir	Remove directory
find	Find files
man	Manual page
apropos	Find commands by subject
whatis	Find subject of command
vi	Editor
exit / logout	Log out of sustem
shutdown	Shut down system
chown	Change file owner
chmod	Change file permissions
ps	Process list
top	List of top processes
vmstat	Cpu useage
mount	Mount filesystem
umount	Unmount filesystem
cat	Concatenate file(s)
dd	Disk to disk copy
less	More (or less)

Filesystem Permissions

Permission	On File	On directory
Read	View file contents	List directory contents (ls)
Write	Alter file contents	Alter directory contents (delete files, create files)
Execute	Run an executable file	Make it the current directory (cd to it

Numeric codes follow octal notation

```
user  group other
4 2 1 4 2 1 4 2 1
r w x r w x r w x
```

For example, `rwxr---x` becomes 741

Linux Class Reference Card

"The Unix Way"

1. Make each program do one thing well. To do a new job, build afresh rather than complicate old programs by adding new features.

2. Expect the output of every program to be the input to another, yet unknown program. Don't clutter output with extraneous information. Avoid stringently columnar or binary input formats. Don't insist on interactive input.

3. Design and build software, even operating systems, to be tried early, ideally within weeks. Don't hesitate to throw away the clumsy parts and rebuild them.

4.Use tools in preference to unskilled help to lighten a programming task, even if you have to detour to build the tools and expect to throw some of them out after you've finished using them.

"Unix Time-Sharing System Forward", The Bell System Technical Journal, Jul-Aug 1978 vol 57 no. 6 part 2, pg 1902